Survivability-oriented Negotiation Algorithms for Multi-domain Virtual Networks

Rashid Mijumbi*, Juan-Luis Gorricho*, Joan Serrat*, Javier Rubio-Loyola† and Ramón Agüero‡
*Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
†CINVESTAV, 87130 Ciudad Victoria, Tamaulipas, Mexico
‡Universidad de Cantabria, 39005 Santander, Spain

***Abstract*—As network virtualisation continues to receive attention, recent proposals have advocated for survivability in network virtualisation environments (NVEs). However, research work within the same area has mainly focused on the single provider environment, leaving network survivability in multi-domain environments largely unexplored. In particular, survivability in heterogeneous physical networks raises questions with regard to the negotiation between competing parties so as to form coalitions for resource provisioning. In this paper, we propose a distributed negotiation algorithm which uses a system of entities to support survivability in a multi-domain NVE. The objective is to make each of the virtual network providers adaptive and dynamic by modelling them with capacity to perform QoS aware resource back-ups and/or restorations for physical link failures.**

Keywords*—Future Internet, network virtualisation, virtual network embedding, distributed systems, autonomic systems.

I. Introduction

Network Virtualisation is a promising approach towards a more adaptive Future Internet. In a NVE, an infrastructure provider (InP) owns, controls and manages physical resources, which may be used, or offered to third-parties, to build custom-tailored virtual networks (VNs). These third-parties could be virtual network operators (VNO) that handle end user attachment, establish, manage and operate VNs; or service providers (SP), who provide services to end users. Depending on the business model, the VNO and SP could be the same entity. While the VNO could directly request for resources from InPs, in general, it is possible to have a virtual network provider (VNP) who acts as a broker between InPs and VNOs. This way, the VNP assembles a VN, according to descriptions given by the VNO, utilising resources from one or more InPs.

In practice, physical networks do not remain operational at all times [1], hence making the provisioning of resources for service restoration an inevitable part of any survivable network resource management approach. Current approaches to survivable virtual network embedding (VNE) [2] have focused on the single InP environment [3], yet an extension from the single to multi domain environments is not trivial since it involves both intra and inter domain link failures [4].

In this paper, we propose a distributed negotiation algorithm based on a multi-entity system to support survivability in a multi-domain NVE. We model each of the InPs and VNPs as an intelligent autonomic negotiating entity [5]. The task of each VNP entity is to achieve survivable embeddings (by ensurig that the VN it assembles has minimal QoS violations resulting from physical resource failures) at lower costs, while each InP entity tries to maximise its profits by keeping resource deployment and pricing strategies private. We begin by extending the VN and InP models [6] to include parameters which define a multi-domain NVE, then present a negotiation protocol finally propose a dynamic pricing model for InP resources.

Since network link failures occur about 10 times more than node failures [3], and given that about 70% of unplanned link failures are single link failures [7], this paper focusses on protecting and restoring single substrate link failures. We however note that any node failure can be considered as a failure of links adjacent to the node [4], and as such, our proposal can be extended to cover multiple link failures, and hence node failures. To the best of our knowledge, this is the first endeavor to propose survivability in independent multi-domain NVEs. Detailed surveys of related work can be found in [2], [3].

The rest of the paper is organised as follows: Section II describes the problem for which the negotiation protocol and dynamic pricing model presented in Section III is proposed. An example workflow of our proposal is presented in Section IV, and we conclude the paper in Section V giving a brief description of the next steps in this work.

II. Problem Description

Link failures can be managed by either provisioning backup resources, or by attempting to perform re-routing upon failures [4]. While it would be more resource efficient to wait for links to fail and thereafter perform re-routing of the affected paths, re-routing schemes can be time consuming since the availability (or not) of resources to support backup links has to be established at fault time [8].

A. Business Model

The business model considered in this paper is shown in Fig. 1. VNOs provide all their requirements for creating VNs to VNPs, and they (VNOs and VNPs) have SLAs with regard to VN provisioning, for example, in terms of virtual network downtime. We consider that the agreements between VNOs and VNPs involve varying penalties for violating QoS, and that the biggest contributors to QoS violation are substrate link failures, which ultimately lead to virtual link failures. Therefore, to guard against high penalties resulting from QoS violation, a VNP takes decisions with regard to backing up of virtual links. The objective of the VNPs is to maximise its profits by minimising both QoS violation penalties as well as the high expenses from resource backup reservations.

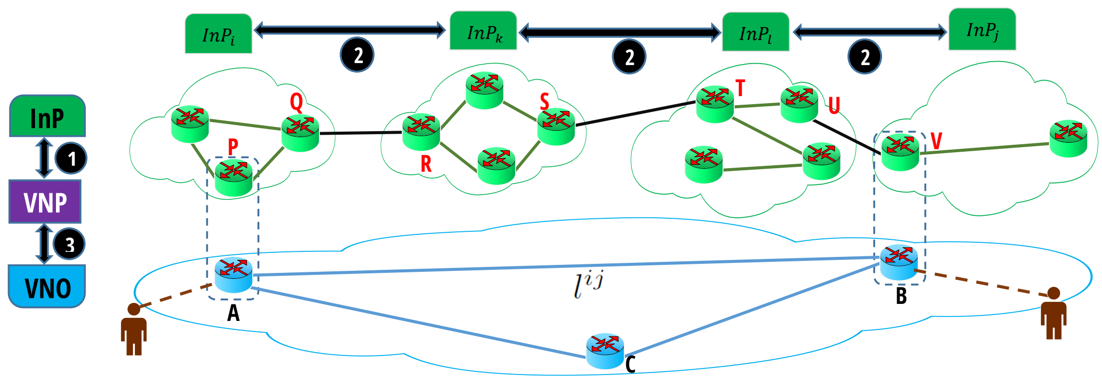

Fig. 1. Multi-Domain Virtual Network Embedding Problem Formulation

B. Virtual Network Embedding (VNE)

Except for the penalties due to failed links, the overall relationship between VNOs and VNPs is well defined in the state of the art e.g. in [9], [6], and mainly involves virtual network modelling and VNE. For this reason, this paper starts after a successful VNE[1]. We concentrate on the interactions between the VNP and InPs after the initial embedding stage, which consist of creating survivable virtual links, by provisioning back-up links for each of the already mapped virtual links, and the negotiation algorithm which provides support to these interactions. The idea is to reserve resources that can be used by virtual links in case of failures in the substrate network. This, however, must be done carefully to avoid that VNPs incur very high costs for resource reservations.

C. Virtual and Substrate Network Modelling

To achieve inter-domain mapping, InPs should make inter-domain connections to InPs that map the two ends of a virtual link. Considering Fig. 1, the two ends A and B of virtual link l^{ij} have been mapped by InPs InP_i and InP_j respectively. For the link l^{ij} to be mapped, all the four InPs must participate in the mapping. Before any InP that is not mapping any of the two end nodes (e.g. InP_k and InP_l) can participate in the mapping, at least an immediate neighbour must have participated in the mapping, or one of the end nodes of the virtual link must have been mapped by its direct neighbour. As an example, for InP_k to connect to virtual node B of virtual link l^{ij}, then InP_l must have a connection to the same via InP_j i.e InP_l should have participated in the mapping. For this reason, in addition to the parameters used to model virtual and substrate networks [11], in this paper each virtual link l^{ij} whose ends belong to InP_i and InP_j has to be characterised by an InP, InP(l_u^{ij}), which performed the most recent connection to the virtual node u of the link l^{ij}. This means that since the mapping of a given link may involve more than one InP, the mapping always starts from one end and ends and the other (or starts from both ends and joins in the middle). As an example, the virtual link l^{ij} in Fig. 1 is mapped onto the multi-domain substrate path PQRSTUV. Assuming that InP_i makes an initial mapping for the substrate path PQ and forwards the mapping request to InP_k, then according to InP_k, the most recent connection to virtual node A was performed by InP_i. This is necessary because before InP_k attempts to add its own intra-domain mapping RS, it should first connect to InP_i to add the inter-domain path QR. The information is used during negotiation to allow InPs that receive link mapping requests to know which InP to contact so as to perform inter-domain mapping, and hence ensure connectivity. This information does not reveal any private information about InP(l_u^{ij}), except that it participated in the resource coalition, and is only known to an adjacent InP. For the scenario in Fig. 1, for InP_k to make a connection to the virtual node B, then InP(l_B^{ij}) must be InP_l. In addition, each virtual link l^{ij} has a length l_x^{ij}, a bandwidth requirement l_b^{ij}, and a QoS value, l_q^{ij} units.

[1]For future evaluations of our proposals, PolyViNE [10] will be used to achieve the initial VNE.

III. Negotiation Protocol

This is a set of rules that governs the interactions between entities, i.e. between InPs and VNPs. The protocol defines the flow of messages both amoung InPs, and between InPs and VNPs. After the VNE step, the VNP is tasked with backing up each substrate link or path that has been used to map any of the virtual links. In what follows, we define the 7 messages that form the proposed negotiation protocol, which in turn are graphically represented in Fig. 2 in a typical negotiation process.

- **Service Request**(InP_i, InP_j, l_b^{ij}, ID, $BlackList$, $Expiry$): A service request (SR) message is sent by either a VNP (to initiate negotiation) or by InP (to forward mapping request) to a given set of InPs to request for mapping of a given virtual link with a unique identification ID. In case it is sent by a VNP, it is the first message in the negotiation process, and initiates the provisioning of the backup for a virtual link. In case it is sent by an InP, this message represents forwarding of a given virtual link backup provisioning. InP_i, InP_j and l_b^{ij} have been defined in Section II-C. $BlackList$ is a set of InPs which cannot participate in the resource coalition. This may be due to policy considerations, or the fact that a given InP has already participated in the mapping, and hence re-sending a SR to this InP would only increase message exchanges. It is worth noting that for privacy reasons, the actual substrate node mapping the end nodes of the link are not revealed during the message exchanges, instead giving the link ID so that the responsible InP can use it to find the

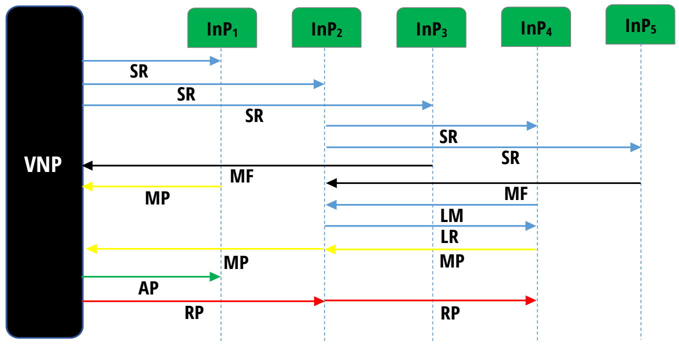

Fig. 2. Message Exchange between InPs and VNP

actual start and/or end node. Finally, each SR has a fixed life time represented by $Expiry$. This ensures that the mapping attempts for a given link can only go on for a given time, saving possibilities of flooding the network with mapping requests that cannot be fulfilled.

- **Mapping Proposal**(ID, $Price$, $NumLinks$, $Loading$): After receiving a SR, an InP attempts to perform a link mapping. If the mapping is successful, the InP replies with a mapping proposal (MP) to the sender, giving details of the mapping such as $NumLinks$ which is the total number of substrate links onto which the virtual link is mapped, (which is a measure of the total expected delay), $Loading$ which is the maximum of the percentage loadings of all of the substrate links participating in the virtual link mapping (which is a measure of possible service invocation failures due to diminished substrate link bandwidth, or substrate link failures due to over loading), and the $Price$, which is the cost of the embedding, per unit bandwidth and time.
- **Reject Proposal**(ID): After receiving a MP, a given InP or VNP may reject it either due to policy violations, or being above the reserve cost expectation, or not the winning bid. In this case, a reject proposal (RP) message is sent.
- **Accept Proposal**(ID, $Price$): If on the other hand a VNP or InP determines that a given MP is the best, or acceptable considering both pricing and policy aspects, an accept proposal (AP) message is sent to the corresponding InP. The $Price$ is specifically useful if the negotiation is between two InPs as this will usually be different (higher) from the one which was proposed in the MP.
- **Link Map**(u, $SNode$, ID): In case of links that need to be mapped inter-domain, a link map (LM) message is sent by an InP to $InP(l_u^{ij})$ that performed the last mapping with respect to the virtual node u, which is at the end of the link ID, specifying the inter domain substrate egress node $SNode$ that it connects to. On reception of this message, the receiving InP determines its node $SNode_x$ onto which u was last mapped, and attempts to create a substrate path between $SNode_x$ and $SNode$. Before the LM message is sent, the sending InP ensures that the inter-domain link connecting them has enough resources to support the virtual link. Using the example of the virtual link l^{ij} in Fig. 1 again, and assuming that InP_i has made an initial mapping for the substrate path PQ and forwarded the mapping request to InP_k, then according to InP_k, the most recent connection to virtual node A was performed by InP_i. Therefore, InP_k sends a **LM**(A, R, l^{ij}). With this, InP_i would already know that this virtual link was mapped onto node Q in its domain, and will therefore attempt to create the inter-domain link QR.
- **Link Result**(ID, $Price$, $Result$): The link result (LR) message is sent in response to a LM, after attempting to make a connection to the egress node of the sending InP. $Result$ is a binary value that is 1 if the mapping is successful, and 0 otherwise. If $Result == 1$, then $Price$ is the cost of the link mapping, which the sending InP should add to its mapping cost.
- **Mapping Failed**(ID): The mapping failed (MF) message is sent by any InP to either an InP or VNP when a mapping cannot be provided either due to policy violations or resource constraint restrictions.

A. *Pricing Model*

In order for InPs to generate proposals in response to a service request, they should be able to determine prices for their resources. We have chosen to use a hybrid pricing function that is based on the logistic function. This pricing model represents a dynamic pricing scheme that is based on the level of resource utilisation for the substrate network, which is restricted at either end by maximum and minimum allowed prices for the substrate resource in question. This pricing model has advantages over the constant pricing model that has been used in most network virtualisation proposals such as [2], [4] and [10], as it does not only allow prices to reflect network loading (hence encouraging better resource utilisation, and minimising network failures from over loading), but also ensures that resources have reserve prices (to cater for minimum fixed costs), and maximum prices to ensure competitiveness. Therefore, we model the price per unit of flow $P(s)$ on a substrate link s as shown in (1).

$$P(s) = l_x^s\left(P_{min}^s + \frac{P_{max}^s - P_{min}^s}{1 + exp\Big(c_1 - c_2u(s)\Big)}\right) \tag{1}$$

where P_{min}^s is the minimum acceptable price for s whose resource utilisation level is $u(s)$ and length l_x^s, and P_{max}^s is the maximum allowed price. c_1 is a constant aimed at shifting the pricing function horizontally (and hence affecting the levels of resource utilisation where the *minimum* and *maximum* prices come into effect), and c_2 is a constant that determines the slope of the pricing function (and hence the rate at which pricing changes from minimum pricing to maximum price). Therefore, the total price C^s that should be paid for all the secondary flows i.e. flows over backup resources f_v^s is given by (2)

$$C^s = \sum_{all f_v^s} f_v^s P(s) \tag{2}$$

The resulting pricing curve is shown in Fig. 3. While in general a purely dynamic environment would benefit if all the parameters (P_{max}^s, P_{min}^s, c_1 and c_2) in (1) are dynamically adapting to resource availability and InP policies, it is out of the scope of the current work to evaluate these possible advantages in NVEs. Therefore, we consider that for any given substrate link, these parameters are fixed. It should also be remarked that the survivability approach in this paper is independent of pricing model, and as such, other pricing models could be used.

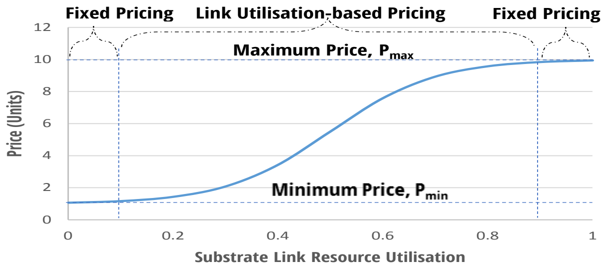

Fig. 3. Substrate Resources Pricing Utility Function

IV. Work Flow

To give an illustration of our proposal, we give an example negotiation work flow based on Fig. 1, Consider that a VNP wants to provision backup resources for the virtual link l^{ij}. We assume that the virtual link l^{ij} has already been mapped, with its two ends A and B being mapped by InPs InP_i and InP_j respectively[2].

The VNP starts by determining an initial set of InPs to which the request can be sent. This initial set of InPs is such that it includes InPs that performed the initial mapping (and/or their direct neighbours) of the virtual link under consideration. For virtual link l^{ij}, the initial set would include the InPs InP_i and InP_j. With the InP set determined, the VNP sends the same service request (request to provision backup resources for a given virtual link) to each of the InPs in the set. The request includes the identity of the InPs that are mapping each end of the virtual link.

On reception of a request from the VNP, a given InP begins by determining if it is able to complete the mapping on its own, i.e. if both ends of the virtual link are mapped with in its domain, and it has enough substrate link resources to provision the link. If the InP can perform the mapping on its own, it creates a proposal using the pricing model in III-A, and sends it to the VNP. However, in the example of Fig. 1, InP_i is not able to complete the mapping on its own since one end of the virtual link is mapped by a different InP. In this case, InP_i would forward the request to (its direct neighbour) InP_k.

Whenever an InP receives a forwarded request from one of his neighbours, it starts by ensuring that the inter-domain link connecting them has enough capacity to support the service being requested. If the inter-domain link does not have this capacity, then, the mapping cannot be completed, and the VNP will be informed about the failure. In our case, this means that InP_i must be able to provision link resources from node P (which maps one end of the virtual link), to node R (in the InP where the request has been forwarded). For instance, these resources could be along the substrate path PQR. At this point, since InP_k already has a connection to the node A of the virtual link (through the path PQR), the request issues from InP_k will include InP_k as the "most recent connection to the virtual node" (see subsection II-C). Therefore, the requests forwarded by InP_k will be a provisioning request for a link starting from InP_k to InP_j. Following a similar procedure, InP_l and InP_k will collaborate to create the connection RST, and finally, InP_l and InP_j will create the final path TUV. At this point, InP_j will send back its cost to InP_l, who would, after adding his own cost forward his proposal to InP_k, and so on, until a final mapping proposal is delivered to the VNP. On reception of a proposal, the VNP may accept or reject it based on its own evaluation.

[2]It is worth noting that while the example in this section only considers a single link, the general process will involve identical processes for each of the virtual links that are part of an embedded VN, and subsequently multiple VNs arriving one at a time.

V. Conclusion

This paper has proposed a distributed and dynamic negotiation protocol that allows for a survivability-aware virtual network embedding in multi-domain environments. We proposed a model of the substrate and virtual networks that allows for this negotiation, a utilisation-based dynamic pricing model for substrate network resources and given an example workflow of how our proposal would function. However, this paper presents only initial ideas of our proposals. We have already developed a strategy VNPs to evaluate and select the best proposals. We have also defined the various attributes e.g. price and QoS that the negotiating parties can combine to form multi-attribute negotiations. We are currently performing simulations to evaluate our proposals to be presented in a subsequent paper.

Acknowledgment

This work was partly funded by project TEC2012-38574-C02-02 from Ministerio de Economia y Competitividad; FLAMINGO, a Network of Excellence project (318488) supported by the European Commission under its Seventh Framework Programme; the CONACYT FOMIX project TAMPS-2012-C35-185768 and CONACYT-CDTI Project 189413.

References

[1] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an ip backbone. In *2nd ACM SIGCOMM Workshop on Internet Measurment*, pages 237–242, 2002.

[2] A. Fischer, J.F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906, Fourth 2013.

[3] A. Khan S. Herker and X. An. Survey on survivable virtual network embedding problem and solutions. In *International Conference on Networking and Services*, ICNS 2013, March 2013.

[4] M. R. Rahman and R. Boutaba. Svne: Survivable virtual network embedding algorithms for network virtualization. *Network and Service Management, IEEE Transactions on*, 10(2):105–118, 2013.

[5] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.

[6] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862 – 876, 2010.

[7] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, Y. Ganjali, and C. Diot. Characterization of Failures in an Operational IP Backbone Network. *Networking, IEEE/ACM Transactions on*, 2008.

[8] L. Huhnkuk and L. Youngho. Toward reliability guarantee vc services in an advance reservation based network resource provisioning system. In *In Proc. of the Eighth International Conference on Systems and Networks Communications*, ICSNC2013, pages 112 – 120, November 2013.

[9] R. Mijumbi, J.L. Gorricho, and J. Serrat. Autonomic resource management in virtual networks. In *Proceedings of the Scalable and Adaptive Internet Solutions (SAIL)*, 2012.

[10] F. Samuel, M. Chowdhury, and R. Boutaba. Polyvine: policy-based virtual network embedding across multiple domains. *Journal of Internet Services and Applications*, 4(1):6, 2013.

[11] R. Mijumbi, J. L. Gorricho, J. Serrat, K. Xu, M. Shen, and K. Yang. A neuro-fuzzy approach to self-management of virtual network resources. *Journal of Expert Systems With Applications*, Sept 2014.